



**NRL/MR/5650--11-9310**

# **On the Potential Use of Evolutionary Algorithms for Electro-Optic System Design**

COLIN C. OLSON

*GTEC, Inc.*

*Crofton, Maryland*

JONATHAN M. NICHOLS

*Optical Techniques Branch*

*Optical Sciences Division*

CARL A. VILLARRUEL

FRANK BUCHOLTZ

*Photonics Technology Branch*

*Optical Sciences Division*

March 25, 2011

# REPORT DOCUMENTATION PAGE

Form Approved  
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instructions, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing this collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden to Department of Defense, Washington Headquarters Services, Directorate for Information Operations and Reports (0704-0188), 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302. Respondents should be aware that notwithstanding any other provision of law, no person shall be subject to any penalty for failing to comply with a collection of information if it does not display a currently valid OMB control number. **PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ADDRESS.**

1. REPORT DATE (DD-MM-YYYY) 25-03-2011	2. REPORT TYPE Memorandum Report	3. DATES COVERED (From - To) 1 Oct 2008 - 30 Sep 2009		
4. TITLE AND SUBTITLE  On the Potential Use of Evolutionary Algorithms for Electro-Optic System Design		5a. CONTRACT NUMBER		
		5b. GRANT NUMBER		
		5c. PROGRAM ELEMENT NUMBER ONR 61135N		
6. AUTHOR(S)  Colin C. Olson,* Jonathan M. Nichols, Carl A. Villarruel, and Frank Bucholtz		5d. PROJECT NUMBER		
		5e. TASK NUMBER		
		5f. WORK UNIT NUMBER WU EW014-02-41-8955		
7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)  Naval Research Laboratory 4555 Overlook Avenue, SW Washington, DC 20375		8. PERFORMING ORGANIZATION REPORT NUMBER  NRL/MR/5650--11-9310		
9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)  Office of Naval Research One Liberty Center 875 North Randolph Street, Suite 1425 Arlington, VA 22203-1995		10. SPONSOR / MONITOR'S ACRONYM(S)  ONR		
		11. SPONSOR / MONITOR'S REPORT NUMBER(S)		
12. DISTRIBUTION / AVAILABILITY STATEMENT  Approved for public release; distribution is unlimited.				
13. SUPPLEMENTARY NOTES  *Global Defense Technology and Systems, Inc., 2200 Defense Highway, Crofton, MD 21114				
14. ABSTRACT  This report discusses the feasibility of employing evolutionary algorithms for the automatic design of electro-optical systems for a wide variety of DoD applications. Evolutionary algorithms (EAs) can be utilized for automatic design provided potential solutions can be encoded mathematically, can be rapidly tested analytically, computationally, or experimentally, and can have their performance quantified mathematically. Importantly, EAs can be employed even when changes in system design lead to abrupt or discontinuous changes in system performance. As an illustration, we use an EA to automatically design a simple lens.				
15. SUBJECT TERMS Evolutionary algorithm      Electro-optic system Automatic system design				
16. SECURITY CLASSIFICATION OF:  a. REPORT Unclassified		17. LIMITATION OF ABSTRACT  SAR	18. NUMBER OF PAGES  21	19a. NAME OF RESPONSIBLE PERSON Frank Bucholtz
b. ABSTRACT Unclassified				19b. TELEPHONE NUMBER (include area code) (202) 767-9342



## Executive Summary

- The purpose of this report is to describe the feasibility of employing evolutionary algorithms (EAs) for automatic design of electro-optic system architectures for a wide variety of DoD applications.
- Evolutionary algorithms can be employed for design provided potential solutions (i.e., designs) can 1) be encoded mathematically, 2) be rapidly tested analytically, computationally, or experimentally, and 3) have their performance quantified by means of an objective function.
- Importantly, EAs can be used when different trial solutions produce abrupt or discontinuous changes in system performance. Traditional methods like “steepest descent” fail in a discontinuous objective space.
- As an illustration, we apply an EA to the problem of automatically designing a simple optical lens using ray tracing and Snell’s law.
- EAs offer the potential to design entire systems from scratch, with no human input other than a specification of constraints and the desired relationship between input and output.

## Introduction

Electro-optic (EO) systems play an integral role in today’s military and are used routinely in land-, sea-, and air-based operations. Examples include night vision displays, ground station instrumentation for target detection and acquisition, submarine periscopes, and electronic warfare (EW) systems. The impressive diversity of EO systems is accompanied by an equally impressive number of variables to be considered in their design. According to Driggers *et al.* there are “easily over 100 parameters to address before the construction of an I<sup>2</sup>R [imaging infrared] or EO sensor” [1]. This complexity makes it impractical to consider all possible device architectures for a given application and the designer must rely on engineering judgment and past experience to narrow the choices.

The design process is essentially a constrained optimization problem: create a system that accomplishes a desired task given limited resources. From this perspective, the human engineer might be considered to be a sophisticated optimization algorithm, albeit, one that is poorly understood and can’t yet be encoded computationally. If design is simply constrained optimization, then it should be possible to program an algorithm capable of searching for the

best solution, that is, automate the design process. The commonly-known, first-order task for optimization routines is the problem of finding the best parameters given a fixed system architecture; what is perhaps less developed is the idea of *optimization algorithms that are capable of designing entire system architectures* in addition to selecting the preferred parameters. In short, is it possible to develop algorithms that are capable of designing human-competitive systems?

The answer to this question appears to be “yes”. Researchers have leveraged the idea of Darwinian natural selection to generate human-competitive designs using evolutionary algorithms. As long as potential solutions can 1) be encoded mathematically, 2) be rapidly tested analytically, computationally, or experimentally, and 3) have their performance quantified via an *objective function*, then EAs can be a valuable design tool. Some more prominent examples include a compact antenna that was flown on a NASA mission and found to outperform previous designs [2] and patentable circuit designs for applications ranging from digital filters to control systems [3]. These human-competitive designs are tantalizing evidence that the design process can indeed be automated and offer the possibility of creative and heretofore unknown solutions to many problems of interest.

Evolutionary algorithms are designed to solve high-dimensional optimization problems for which a well-defined global maximum (or minimum) is otherwise difficult to find and for which an exhaustive search is not possible. Difficult optimization problems include those that are characterized by multiple optima, discontinuities, constant-valued regions of the objective space or some combination thereof. Using traditional gradient-ascent optimization routines on objective spaces with these characteristics can yield poor results if, for example, the routine converges to a local rather than a global optimum. EAs are designed to reduce the possibility of this occurrence and are capable of dealing with problems like discontinuities.

We discuss how the objective space corresponding to system architectures for a given design problem is likely to have some of these difficult characteristics and why EAs should be considered for the automated design process. We begin with a discussion of evolutionary algorithms, then provide an example of how we have used EAs to design a lens using ray tracing and Snell’s Law, and finally, we close with example applications that might benefit from this approach.

## Background

Optimization problems whose solution spaces are discontinuous, high-dimensional, and/or characterized by many local optima are often encountered in engineering practice. Traditional gradient-based optimization methods have difficulties with such spaces because gradients are undefined across discontinuities and because it is easy for such

algorithms to converge to local optima without ever finding the global optimum. This erroneous convergence occurs because gradient-based methods utilize local information about solution space topology to make decisions about how the parameters should be perturbed in search of the optimal solution. Thus, they can converge to local optima or become stuck in regions of the solution space where changes in the parameters yield no change in the objective function. As an example, the solution space shown in Figure 1 is a difficult one for any optimization routine and illustrates many of the features that would make a gradient-based search impractical.

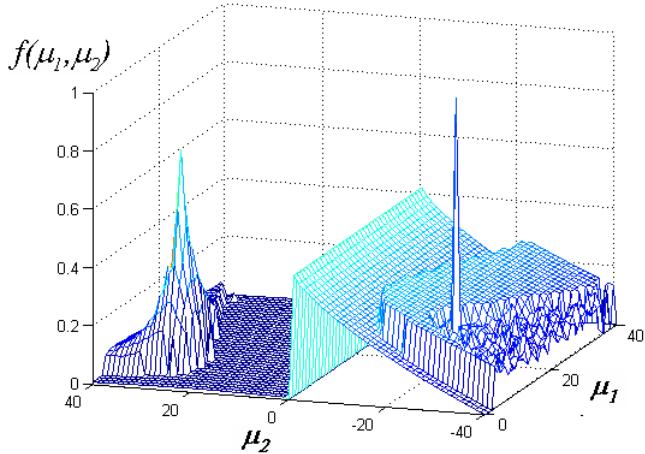


FIG. 1: A complex solution space illustrating discontinuities, slowly varying or constant regions, and multiple local optima. It is assumed that the objective function should be maximized.

In addition to becoming stuck in suboptimal portions of the solution space or prematurely converging to a local optimum, gradient-based methods require a gradient calculation—a problem if the objective function is not explicitly defined. In such cases, the derivative of the objective function with respect to each of the parameters must be calculated numerically using finite differences. This requirement can be problematic for three reasons: (1) non-smooth (and especially discontinuous) solution spaces will confound the algorithm, (2) regions of the parameter space where the local gradient is small can trap the algorithm, and (3) high computational expense.

For example, Newton-type optimization routines make use of second-derivative information (the Hessian) in order to avoid becoming stuck at non-optimal stationary points (e.g., saddle points) and to speed the convergence process [4]. Unfortunately, numerical calculation of second derivatives necessitates  $O(n^2)$  calls to the objective function (where  $n$  is the number of parameters) for each step of the algorithm. In addition, “extreme care must be exercised in choosing the finite-difference interval” [4], which, coupled with the computational expense, makes such algorithms “impractical” if analytical derivatives or special circumstances (like a sparse Hessian) are not available [4]. Instead,

quasi-Newton-type methods are recommended as they only require first derivatives. Yet, even with smooth functions, these methods will converge to saddle points, suffer difficulties with the choice of step size, require  $n$  or  $2n$  function calls for each step, and “Even with the best possible implementation...may fail to make any progress in a region where [the magnitude of the gradient] is small but non-zero” [4]. If the objective function is non-smooth, then even small step sizes in the derivative approximation can be “wildly inaccurate” and such methods can fail [5].

Non-smooth objective functions can be optimized using direct-search optimization techniques [5], which do not require derivatives in order to make decisions about where to move in the solution space. They rely instead on calls to the objective function and choose a path through the solution space based on direct comparisons between objective function values. Although less efficient for smooth solution spaces where the gradient can successfully be determined, they are often the only choice for non-smooth problems. Yet, such techniques can still prematurely converge to local optima if there is no possibility of (temporarily) accepting poorer solutions. Thus, we can turn to stochastic direct search algorithms such as simulated annealing [6] or evolutionary algorithms [7] as potential means of improving the possibility that a global optimum will be found.

Simulated annealing moves through the solution space by perturbing a single solution and accepting moves that improve the solution. There is, however, a non-zero probability that poorer solutions will be accepted, which allows the algorithm to avoid becoming trapped at local optima. This probability is decreased according to a “cooling schedule” over the course of the optimization, which allows the algorithm to eventually settle to a given optimum. Alternatively, evolutionary algorithms distribute a population of solutions throughout the parameter space and employ principles of Darwinian evolution to exchange information between the best individuals and improve the population over many generations. The use of a population of potential solutions and the possibility that some solutions will “mutate” at random allows an EA to avoid becoming stuck at local optima. We favor EAs because employing a population of solutions reduces the possibility that a particularly poor initial solution (for simulated annealing) will lead to a local optimum. In addition, employing a population of solutions at each generation makes EAs well-suited for parallelization.

The fact that EAs are more likely to converge to a global optimum makes them useful even if the objective function is smooth and continuous. Their ability to deal with non-smooth functions makes them particularly important when designing complicated engineering systems. This is because such systems typically require computational or experimental evaluation for each parameter setting. This lack of an analytical description matters because reliable derivative information can be hard to extract from such optimization problems. In fact, direct search techniques

were originally developed specifically to handle optimizations involving experiments and, concerning computational simulation, “It is widely appreciated in the simulation-based optimization community that the results of complex simulations...may fail to have the level of precision necessary for a reliable finite-difference approximation to the gradient...” [5]. Environmental stochasticity or numerical noise can lead to non-smooth behavior which means that the lack of an analytical description for the objective function can necessitate the use of direct search optimization routines. Additionally, complex design problems are likely to have many “good” but locally optimal solutions, which further indicates the use of global optimizers such as EAs in the design process. Figure 2 provides an example of how a population of initial solutions might converge to the global optimum for a multi-modal objective function.

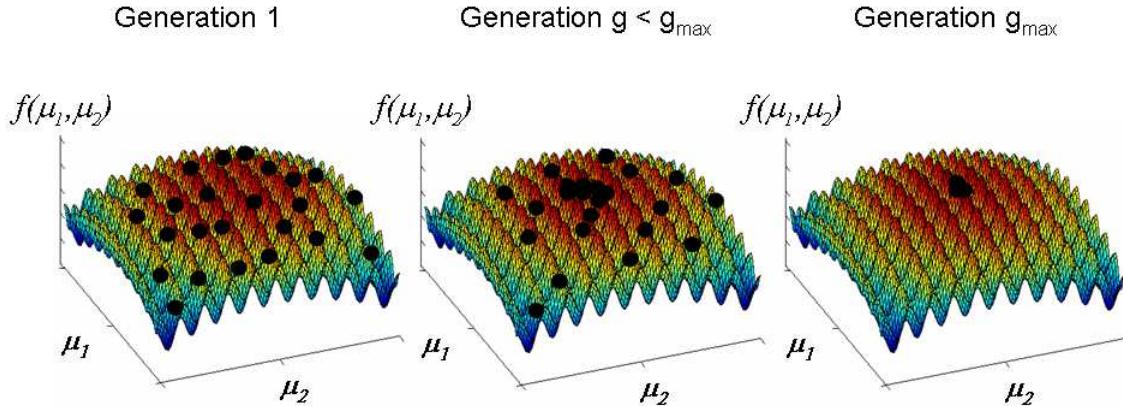


FIG. 2: Hypothetical progression of a population of solutions through a multi-modal solution space with a global optimum.

Evolutionary algorithms can also be used to optimize nonnumerical functions or functions where the derivative is hard to define. An example of the former is given in [5] as a problem of hearing aid design. Noise reduction, phonemic compression, and spectral enhancement were the control parameters, while the goal was to improve listening comfort and speech intelligibility based on a human subject’s evaluation of the settings. Similar problems include the use of EAs to breed art that is pleasing to the human eye.

As an example of a function where the derivative is hard to define, imagine a design problem where the goal is to develop a high-pass analog filter circuit and where the objective function is defined by typical filter characteristics like cutoff frequency and some minimum passband ripple. The derivative of this objective function with respect to changes in circuit topology does not exist because a derivative requires perturbations of the design parameters in a specific direction. It is not possible to define a direction nor a perturbation to the circuit when circuit topology is a parameter. Topological changes to a circuit, like swapping the location of a resistor and a capacitor, are not continuous, ordered operations; the elements have or have not been switched and choosing one over the other is not inherently a larger or

smaller perturbation in some direction. This is an example of a *combinatorial optimization* problem where the various circuit topologies are part of a discrete, factorially large configuration space [8]. Topological changes will lead to large jumps in the circuit design space and, correspondingly, the objective function will be characterized by discontinuities and local optima.

In addition, the objective function cannot be written in closed form as a function of the design parameters. We have no choice but to test each potential circuit computationally or experimentally and we cannot obtain derivative information with which to direct a gradient-based routine. Furthermore, we know little about the nature of the objective function and even the possibility that it is characterized by local optima makes it advantageous to employ evolutionary algorithms to search the configuration space.

## Optical Example: Simple Lens Design Via EA

We provide a brief example to illustrate the design capabilities of an EA by designing a lens using only ray tracing and Snell's law. This is not an especially difficult task, and more complicated optical devices have been designed using EAs [9], but we find it to be a useful pedagogical example. We seek to develop a lens with uniform index of refraction that is capable of focusing a distribution of collimated light rays to a specified focus. Thus, we require some means of parameterizing the lens surface such that we can be assured that the true solution is contained in the space of possible solutions. A 4th-order polynomial will do for this demonstration, but other forms of parameterization (e.g., Bezier curves, higher-order polynomials) could also be used. We choose 4th-order for the sake of simplicity. The upper surface of the lens is defined as  $y_U(x) = a_0 + a_1x + a_2x^2 + a_3x^3 + a_4x^4$  and the lower surface of the lens is simply the mirror of the upper surface,  $y_L = -y_U$  (Figure 3).

In addition, we need an objective function that quantifies how well a given lens solution focuses rays to the desired focus. This requires simulating the effect of each proposed lens surface on a collection of collimated rays and quantifying how close each ray comes to the desired focus. We begin by limiting the domain of  $x$  to the interval  $[-1, 1]$  and choose a uniformly distributed set of values,  $X$ , from the interval. Given a vector of polynomial coefficients,  $\mathbf{a} = [a_0 \ a_1 \ a_2 \ a_3 \ a_4]$ , we carry out a ray tracing operation on each element of  $X$  by first assuming that a given ray vertically intersects the lens surface at the point  $y_U(x_i), x_i \in X$ , and refracts according to Snell's law

$$n_1 \sin(\theta_1) = n_2 \sin(\theta_2), \quad (1)$$

where  $n_1$  and  $n_2$  are the indices of refraction of the medium above and below some surface, respectively, and the

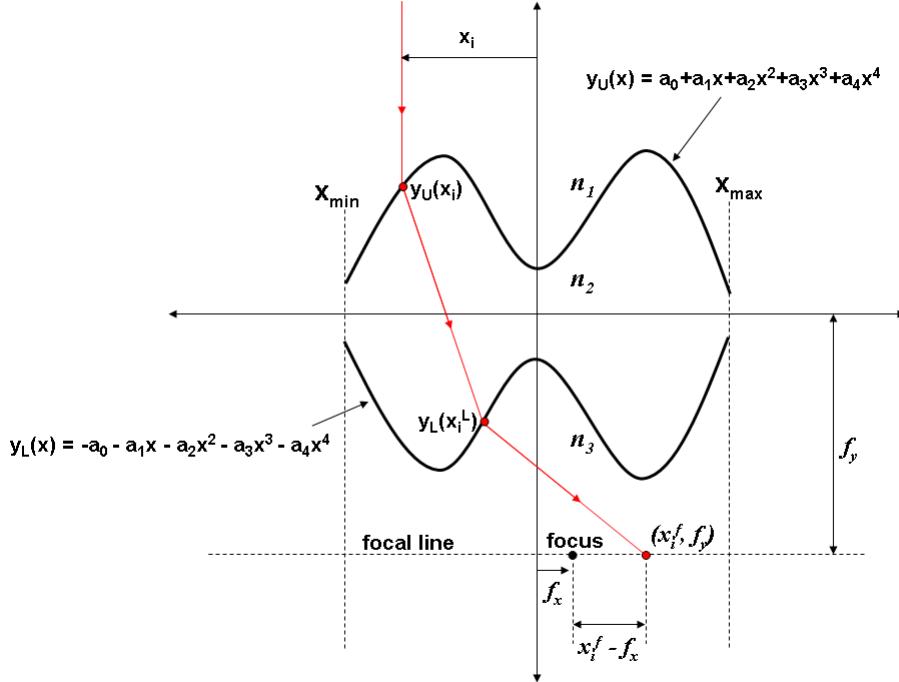


FIG. 3: A two-dimensional lens in the plane where the upper surface is described by a 4th-order polynomial and the lower surface is a mirror of the upper surface.

angles  $\theta_1$  and  $\theta_2$  are the corresponding angles between the ray and the normal to the lens surface before and after refraction (Figure 4).

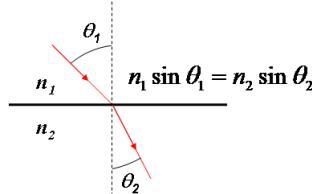


FIG. 4: Illustration of Snell's law where  $n_2 > n_1$ . The angle  $\theta_1$  is measured between the incident ray and the surface normal and  $\theta_2$  is the angle between the refracted ray and the surface normal.

A refracted ray might then intersect the second surface at  $y_L(x_i^L)$ , refract again according to (1) and then (possibly) intersect the centerline (y-axis) at a point  $(0, y_i)$  and, ultimately, intersect the focal line at the point  $(x_i^f, f_y)$ , where  $f_y$  is the y-coordinate of the desired focus. We can determine how well the surface corresponding to  $\mathbf{a}$  has directed the rays to the focus by calculating the Euclidean distance between  $x_i^f$  and  $f_x$ , that is,  $\sum_i (x_i^f - f_x)^2$  where  $f_x$  is the x-coordinate of the desired focus. In this study, the focal point is not offset from the centerline of the lens so  $f_x = 0$ . In essence, the goal of the optimization is to produce a lens that minimizes the spot size under the simplifying

assumptions of ray optics.

Each ray is traced by means of Eq. (1) and simple geometry. First, the derivative of the upper surface is calculated at the current point of interest,  $y_U(x_i)$ , which determines the orientation of, and normal to, the surface at that point. Using Snell's Law, the angle of the ray relative to the surface normal and ultimately the y-axis can be determined. Using basic trigonometry, we extract the slope,  $m_1$ , of a line that passes through that point. Given the equation of the line passing through the point  $y_U(x_i)$ , it then becomes a matter of determining the intersection of that line with the lower surface,  $y_L$ , which requires the roots of a 4th-order polynomial. For rays that intersect the lower surface, there will only be one real root that is within the domain of interest,  $[-1, 1]$ .

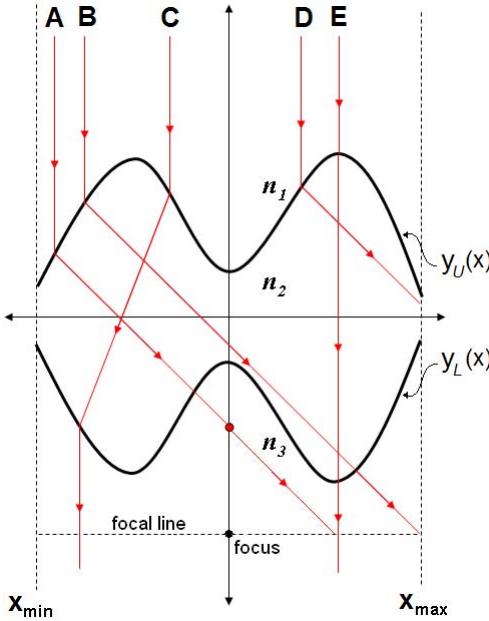


FIG. 5: Illustration of some ray trace possibilities. Trace A passes through  $y_L$  and intersects the centerline. Traces B and C pass through  $y_L$  but miss the centerline. Trace D fails to intersect  $y_L$  within the domain of interest. Trace E passes directly through the lens with no refraction because the derivative of the surface at that point is zero on both surfaces.

For a given ray, if an intersection occurs with the second surface, Snell's law is applied again to determine the slope of the line that intersects the surface  $y_L$  at the point  $x_i^L$ . Given this line, we find its intersection with the y-axis. If the intersection is negative and less than the minimum of  $y_L$  (trace A in Figure 5), then the ray is further traced to determine the x-coordinate of its intersection with the focal line. If the intersection with the centerline is negative but greater than the minimum of  $y_L$ , then the ray has either been refracted away from the y-axis (trace B) or is a ray that intersects  $y_L$  a second time (not shown); either case is penalized in the objective function. If the intersection with the centerline is positive, then the ray has been refracted away from the y-axis (trace C), there will be no intersection,

and this fact is noted for penalty in the objective function. Finally, it is possible a given ray may never intersect the lower surface within the region of interest or at all (trace D) and will also incur a penalty.

Other paths are possible for a given ray. For example, if  $y_L = -y_U$ , a ray will pass directly through the lens with no refraction at either surface if it intersects the first surface at a point where the slope is zero (trace E). Another possibility (not shown) is that the ray will reflect off the second surface if the angle of intersection exceeds the critical angle,  $\theta_c$ , determined by setting  $\theta_2 = 90^\circ$  in (1) and solving for  $\theta_1$ . We do not allow for repeated reflections and penalize any rays that exceed the critical angle at the second surface.

Assuming all ray trace possibilities are properly handled, it is possible to determine whether and where each ray will cross the centerline and ultimately the focal line. A reasonable objective function of a given lens solution is

$$c_{\mathbf{a}} = \sum_{i=1}^{N-N_e-N_r-N_m} (x_i^f - f_x)^2 + \alpha N_e + \beta N_r + \gamma N_m \quad (2)$$

where  $N$  is the cardinality of  $X$ ,  $N_e$  is the number of rays that escape the lens or never intersect  $y_L$ ,  $N_m$  is the number of rays that intersect  $y_L$  but never intersect the centerline,  $N_r$  is the number of rays that reflect off  $y_L$ , and  $\alpha$ ,  $\beta$ , and  $\gamma$  are used, respectively, to weight the associated penalty terms. We note that the selection of these penalty terms only aids with convergence rate and will have no effect on which parameters are eventually found to be optimal. Thus, we seek a vector of polynomial coefficients,  $\mathbf{a}$ , that minimizes Eq. (2). Finally, although we have selected the objective function in Eqn. (2) as a cost to be minimized, it could easily be a fitness to be maximized if the inverse is taken.

We consider rays that fail to pass through  $y_L$  to be an even less desirable condition than rays that pass through  $y_L$  but fail to reach the centerline. Reflected rays are considered to be better than rays that fail to reach  $y_L$  but worse than rays that pass through  $y_L$ . Thus, we set  $\alpha = 20$ ,  $\beta = 15$ , and  $\gamma = 10$  to reflect that opinion. Again, this weighting simply aids with convergence and will not affect the final outcome. In fact, each weighting can be set to zero and the algorithm will still converge to the same result but will require more iterations to do so. Given such a weighting for the cost function, we see how the algorithm progresses through the solution space.

Figure 6 illustrates the evolution of an appropriate lens with differential evolution [10]. Initially, most of the rays initiating from the elements of  $X$  fail to even reach  $y_L$  and the next 100 generations are spent producing a lens surface where most of the rays pass through the lens surfaces and reach the focal line. The following 100 generations lead to a symmetric lens where all the odd polynomial terms have been forced to zero as shown in Figure 7. By generation 200 most of the rays are converging near the desired focal point. The final generations are spent forcing each ray to converge ever closer to the focus.

This example illustrates the capability of an EA to design a lens using only first principles. No information was explicitly provided to guide the algorithm into choosing a quadratic surface but a quadratic surface did have to be one of the solutions available in the space of all possible solutions. Absent the ability to produce the appropriate surface (e.g., if the surface was constrained to be trigonometric), the algorithm would be forced to produce a best approximation.

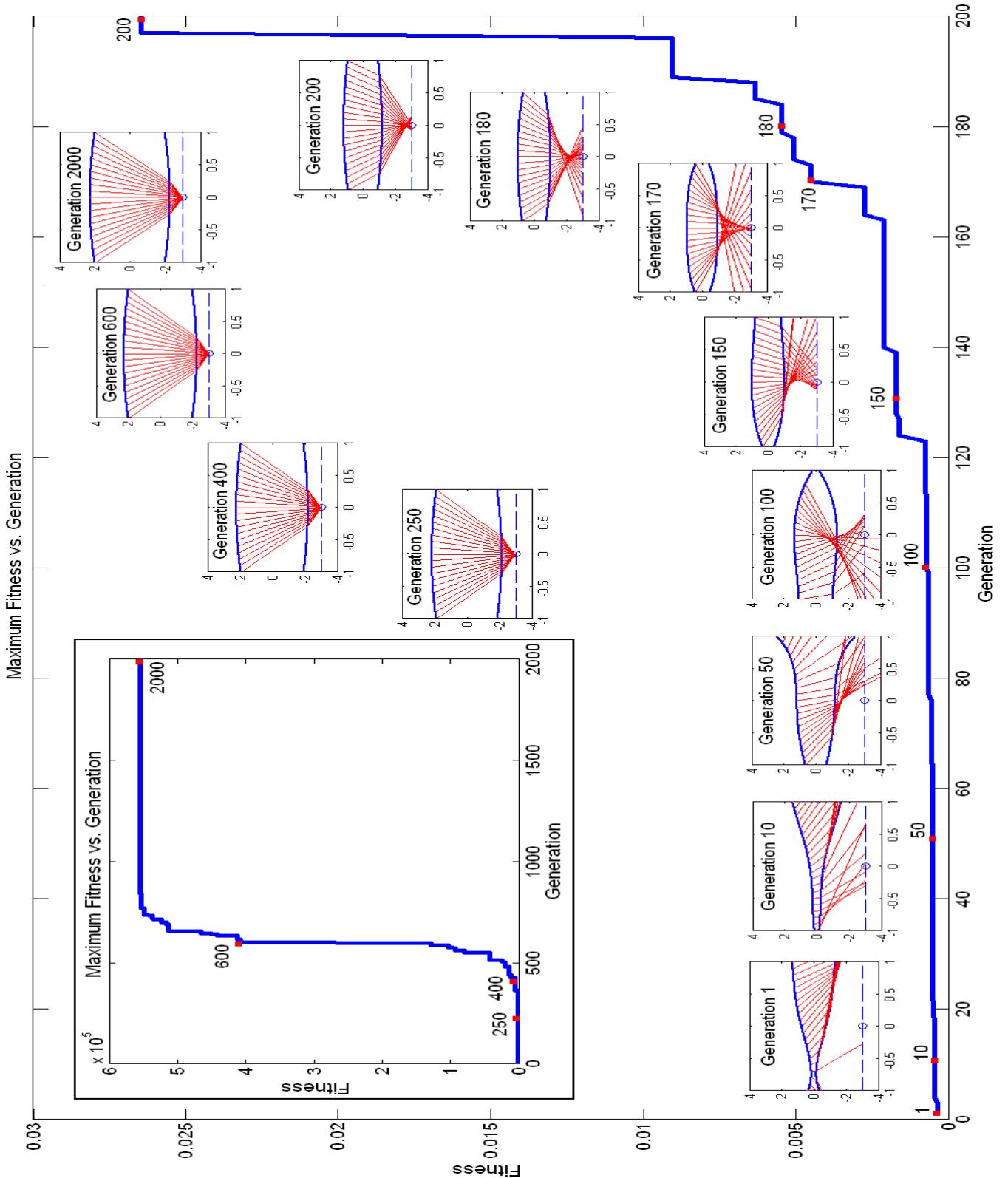


FIG. 6: Fitness ( $1/c_{\mathbf{a}}$ ) vs. generation for the lens design problem with  $n_1 = 1$  (air),  $n_2 = 1.468$  (glass),  $n_3 = 1$ . The upper surface is defined as a 4th-order polynomial and the lower surface is the negative of the upper surface. The optimization seeks the coefficients  $\mathbf{a}$  that best focus the rays to the desired focus. The inset plot shows the entire progression of fitness vs. generation, while the main plot shows the first 200 generations. Each lens sub-image illustrates the best member of the population at that generation.

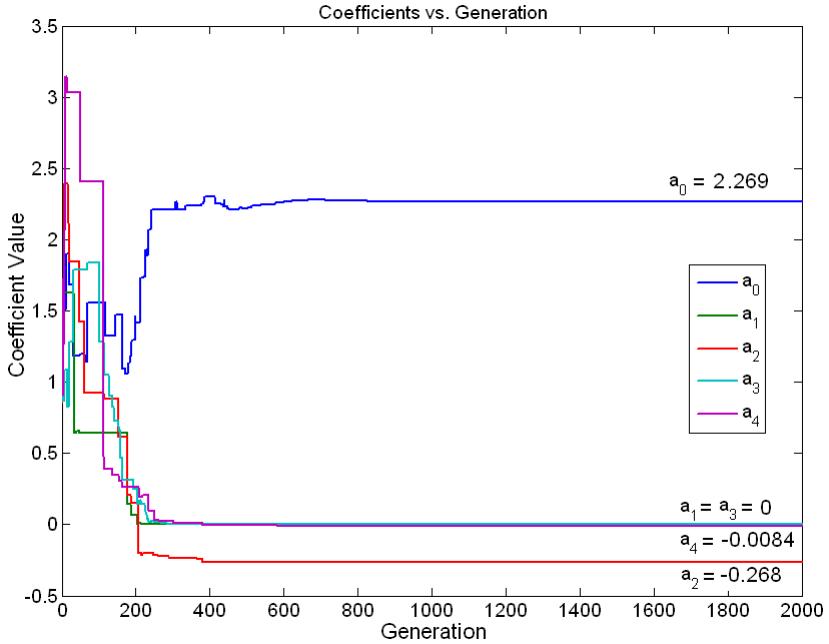


FIG. 7: Parameters of the upper lens surface corresponding to the best solution from each generation. The odd terms are forced to zero as the lens must be symmetric about the y-axis for best performance. The lower surface parameters are the negatives of the upper surface parameters.

Prior knowledge of ray optics allows us to predict that any lens with a focus that resides on the centerline ( $f_x = 0$ ) will require symmetry about the centerline for optimal performance. If we imagine no prior understanding of the problem, the fact that repeated optimization runs force the odd polynomial terms to approach zero would provide evidence that symmetry is required. Thus, given the requisite symmetry of the lens surfaces about the y-axis, we abandon the odd polynomial terms and use only even terms for the optimizations that follow. Several optimization runs composed of 50-member populations were run for 4000 generations under various constraints. Ten optimization runs are performed for each type of optimization. Each of the runs converges to a similar solution for a given set of constraints, the best of which are shown in Figures 8-11. We begin with the simplest optimization which is simply an even quadratic polynomial composed of the  $a_0$  and  $a_2$  terms where symmetry about the x-axis is enforced by  $b_0 = -a_0$  and  $b_2 = -a_2$ , where we have introduced  $\mathbf{b}$  to represent the polynomial coefficients for the lower surface. We perform the optimization for the case of air-glass-air as well as for air-glass-water ( $n_3 = 1.333$ ). The best solutions corresponding to the best optimization run for each case are shown in Figure 8.

The next optimization retains the restriction of an even quadratic polynomial but does not enforce symmetry between the upper and lower surfaces. The  $a_0$  and  $b_0$  terms are, however, forced to remain linked because the focal distance is defined relative to the center of the lens. As long as  $b_0 = -a_0$ , the thickness of the lens will remain centered

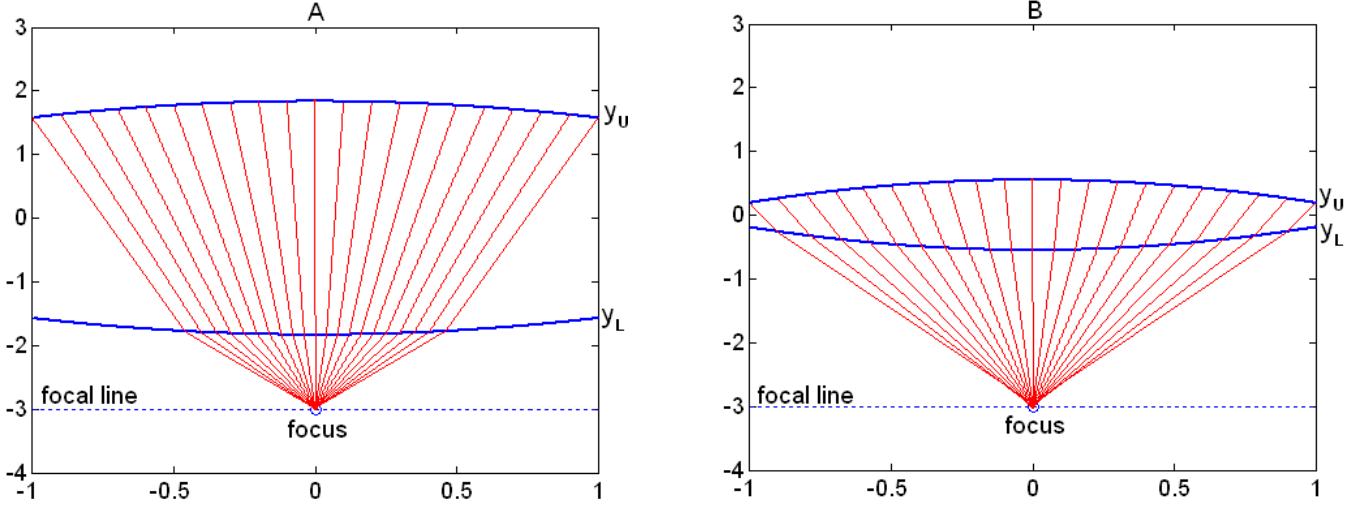


FIG. 8: Best lens solutions for (A) air-glass-air and (B) air-glass-water. Symmetry enforced about x-axis, quadratic surfaces. Fitness for (A) is 274 while that for (B) is 22.

about the origin and the focal distance remains constant for all values of  $a_2$  and  $b_2$ . The results of the best of ten optimizations for each case are shown in Figure 9.

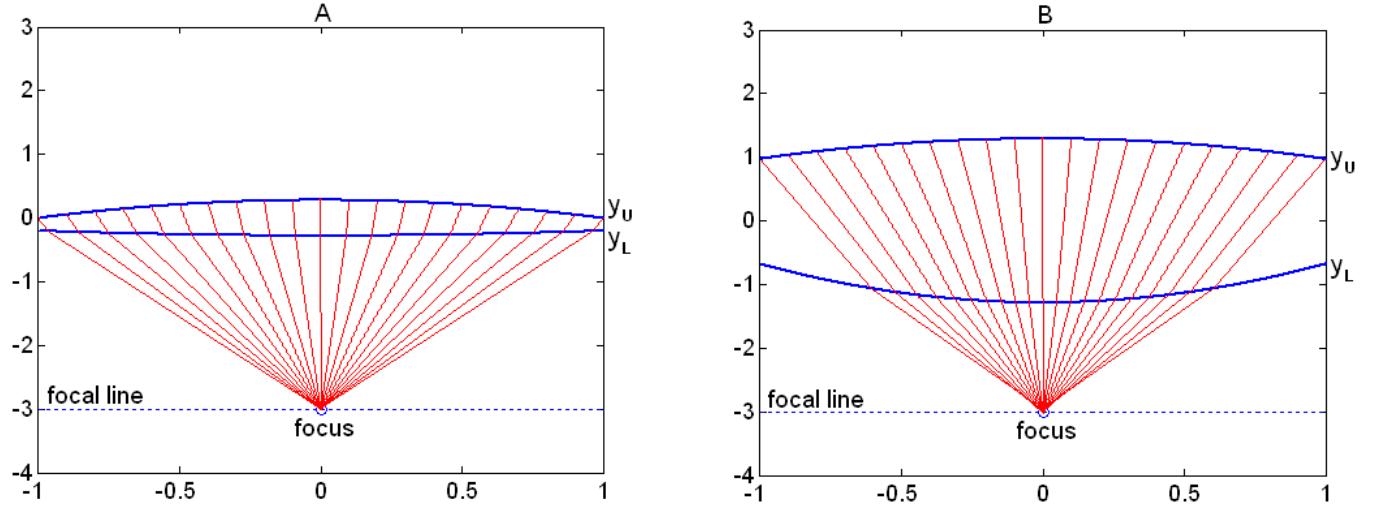


FIG. 9: Best lens solutions for (A) air-glass-air and (B) air-glass-water. Symmetry about x-axis is not enforced but surfaces are still even quadratic polynomials. Fitness for (A) is 608 while that for (B) is 24.

The third optimization enforces symmetry between the upper and lower surfaces but allows a quartic term; results are given in Figure 10. We note that the maximum fitness for this optimization is the same as the maximum fitness for the optimization that included the odd polynomial terms shown in Figure 6. The most general optimization considered allows each surface to vary independently (save for the restriction  $b_0 = -a_0$ ) with a quartic term included. The best lens produced by 10 separate optimizations is shown in Figure 11. We note that the second surface of the

asymmetric quartic lens only affects the path of the rays over a small subsection of the lens, therefore, the negative sloping wings of the second surface have no effect on the solution.

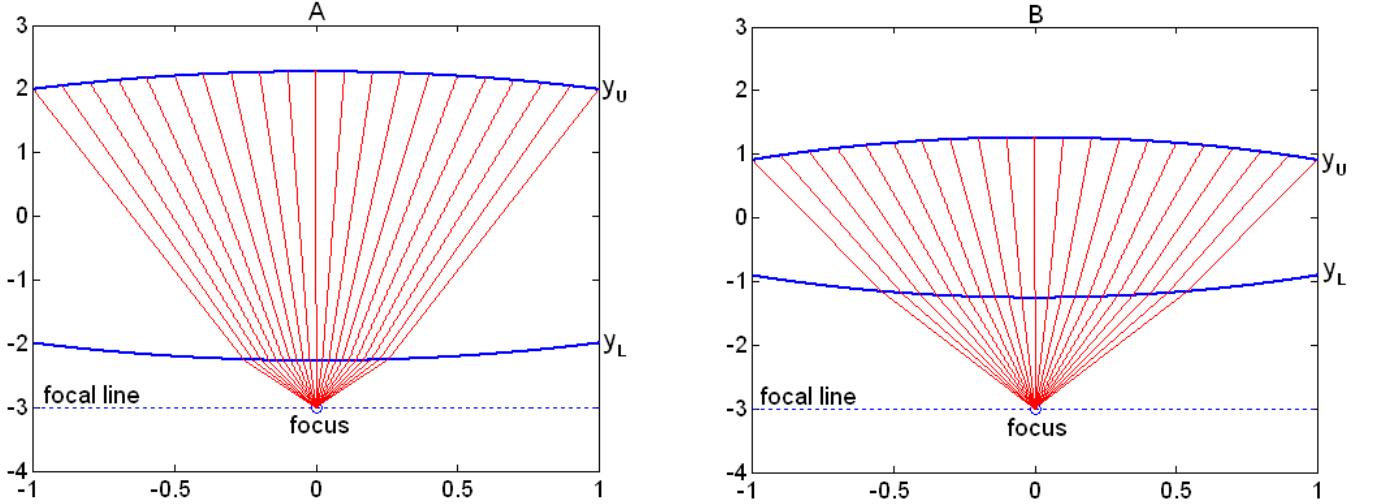


FIG. 10: Best lens solutions for (A) air-glass-air and (B) air-glass-water. Symmetry about x-axis is enforced and a quartic term is added to the surface polynomial. Fitness for (A) is 5.5e5 (note the equivalence with the maximum fitness in Figure 6) while that for (B) is 2.7e4.

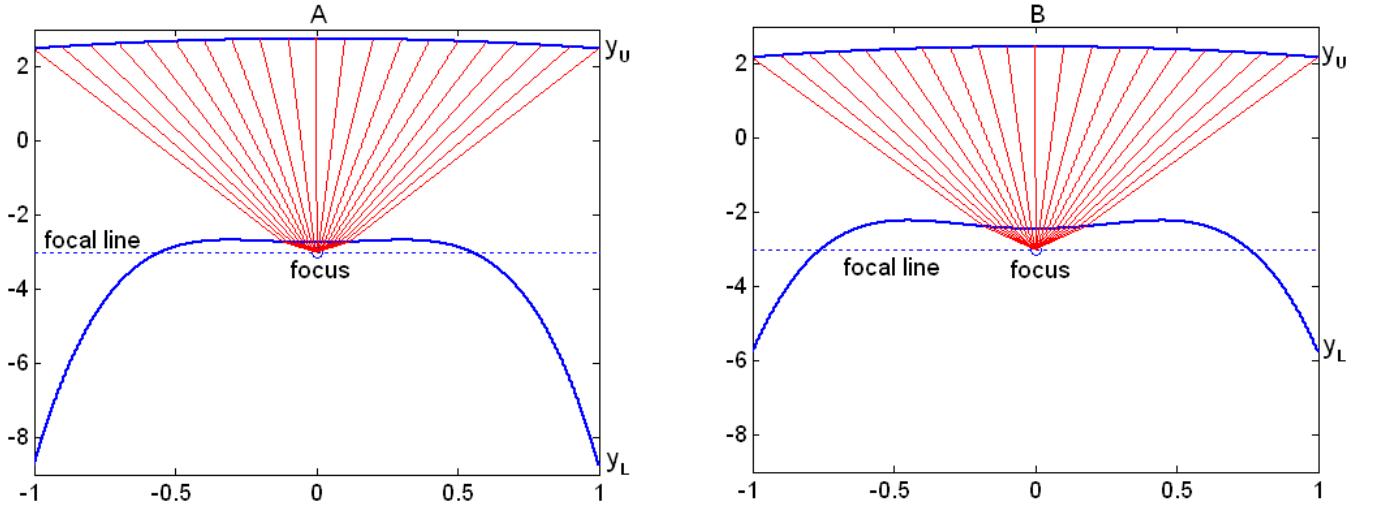


FIG. 11: Best lens solutions for (A) air-glass-air and (B) air-glass-water. Symmetry about x-axis is not enforced and a quartic term is added to the surface polynomial. Fitness for (A) is 2.3e7 while that for (B) is 3.8e6.

This sequence of increasingly general optimizations demonstrates the importance of asymmetry between the two lens surfaces and the quartic term in lens design as predicted by our ray optics model. We have not explicitly shown the relationship between fitness and spot size (as naively measured by the distance between the two most widely separated rays around the focus), but for comparison, that distance decreases by more than an order of magnitude

for the air-glass-air case when the fitness increases from  $5.5e5$  to  $2.3e7$  (Figures 10A and 11A). We stress that these example optimizations are meant to illustrate the utility of an EA for design in an optics application. We do not seek to provide guidance in lens design, as the assumption of ray optics glosses over the many aberrations arising from, among other things, phase and edge effects.

As a final point, we provide an example of the fitness space in Figure 12 (log scale). We start with the best solution discovered from the full quartic optimization (including odd terms) whose results are shown in Figure 6. The  $a_2$  and  $a_3$  terms are then each systematically varied over  $[-1,1]$  with a step size of 0.01 with  $b_2 = -a_2$  and  $b_3 = -a_3$  enforced while the other parameters are held fixed at the optimal values. At each parameter value the ray optics model is run and the fitness of the lens encoded by those parameters is calculated. The peak of the fitness function should occur at  $a_2 = -0.268, a_3 = 0$  as shown. This is a good example of a parameter space that should be searched with a direct-search optimization technique such as an evolutionary algorithm.

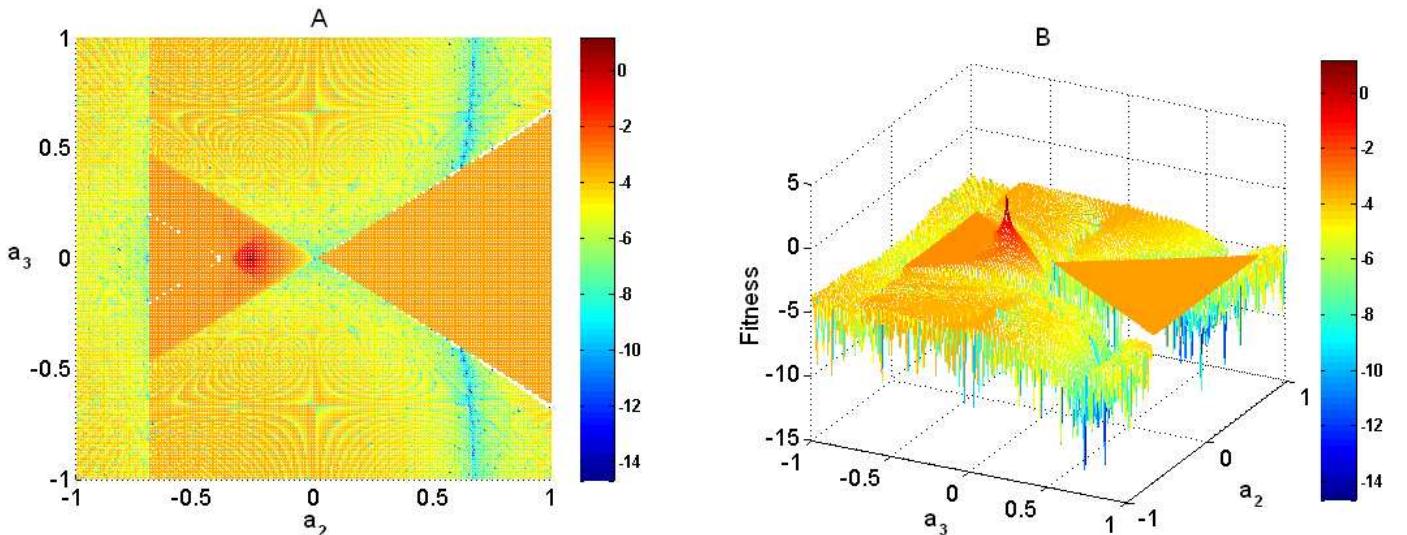


FIG. 12: Fitness (log scale) versus  $a_2$  and  $a_3$  with symmetry enforced. The white spots correspond to zero values. (A) is a top view of the three-quarter view in (B).

## Applications

A tendency to avoid becoming trapped in local optima and the ability to deal with non-smooth functions make EAs excellent tools for designing complicated engineering systems; a reality which has spawned the field of evolutionary-based design. A well-known example from this field is the compact antenna engineered by an EA for use aboard a NASA satellite [2]. The device was found to outperform current human-engineered designs and is a good example of the power of natural selection for finding solutions to constrained problems. Similarly, Koza has used Genetic

Programming (GP) to create analog electronic circuits that outperform modern human-engineered designs. Overall, at least 36 human-competitive results ranging from sorting programs to control circuits have been produced with GP [3]. The design of circuits proceeds by including electrical components such as voltage sources, current sources, resistors, capacitors, transistors, and so forth as elements in a connected graph that describes a circuit. The *topology* of the circuit along with its components can be manipulated by the EA and each generated circuit is tested computationally and its performance compared to a desired standard. Using this framework, the algorithm is capable of starting from a “primordial soup” of components and ending with circuits such as PID controllers, cubing circuits, and various filter types (e.g., high- or low-pass) [11; 12]. Optimizing the elements of a preconceived circuit is useful enough, building the circuit topology from scratch with no human input beyond specification of the desired behavior is an *entirely* different capability with profound implications.

If electronic circuits can be evolved in this manner, there is every reason to believe that electro-optic systems could be similarly designed. As long as each topology can be simulated numerically or experimentally, we believe that an EA can combine, for example, RF amplifiers, splitters, filters, and phase shifters with optical couplers, sources, Mach-Zehnder modulators (MZMs), and Erbium-doped fiber amplifiers (EDFAs) into useful RF/EO systems. In general, an EA could be tasked with designing systems that improve, for example, gain, bandwidth, noise figure, or linearity under constraints such as size, weight, power, and environmental susceptibility.

We are proposing more than simple optimization of the parameters that govern a fixed system architecture. We would like to allow an EA to design the system with no *a priori* constraints on device topology. System performance is acutely architecture-dependent, and the optimal architecture is unknown *a priori*. For example, consider a low-noise downconverting link. Figure 13 shows two separate architectures for the downconverting link as described in [13; 14].

Components for this system include a laser, EDFAs, photodiodes, couplers, and MZMs. The goal is to arrange those components to produce a system that downconverts with minimized noise figure and maximized DC photocurrent under constraints on input optical power and modulation efficiency. We propose to allow an EA to search for architectures that yield results that are competitive with these human-engineered systems.

Potential impact areas include: RF/EO devices, systems, and photonic links, arbitrary waveform generation, array architecture for imaging, and adaptable receivers. More specifically, an EA could be used in the design of nonlinear components and operations for filtering and channelization, high-isolation Tx/Rx, RF waveforms for maximizing linearity, optical pulse shape for lowest attenuation and dispersion in media, and passive microwave imaging to name a few. Modern advances in materials and fabrication techniques allow for a wide variety of devices and architectures

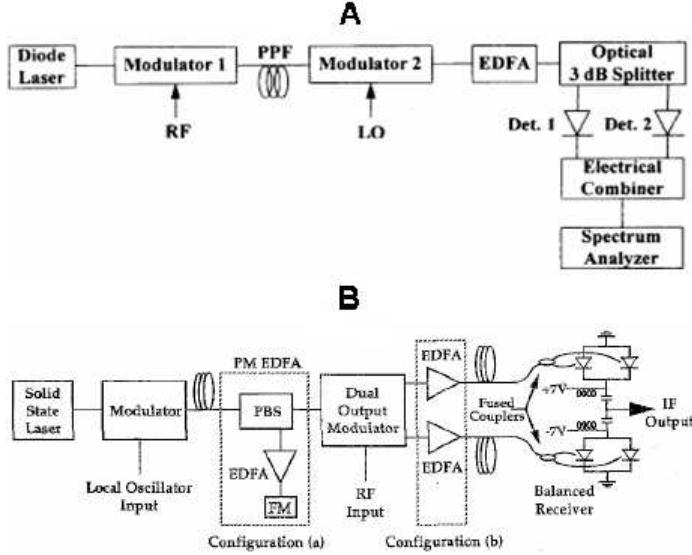


FIG. 13: Two examples of different architectures for a low-noise downconverting link. System A is taken from [13] while system B is from [14].

and the creative capability of an EA in this domain could lead to improved DoD technologies.

The primary challenge will be the development of a technique that allows efficient encoding of topology and components such that an EA can efficiently search the application design space. Koza has illustrated this capability using GP to operate on functions that encode the functionality and interconnectivity of electronic circuit components. The technique is designed to couple into a specific program that simulates the operation of the circuits once they have been designed. An NRL implementation could leverage such capabilities with proper modification to suit RF/EO architectures. Existing EAs such as Gene Expression Programming [15] allow the symbolic manipulation of systems of equations and could be modified to accommodate the demands of architecture design.

## Summary

In the design of RF/EO and EO/IR architectures it is conceivable that the evolutionary algorithm could produce heretofore unknown human-competitive architectures provided the solution space is not overly constrained. Evolutionary algorithms can optimize objective functions that are characterized by multiple local optima, are non-smooth, are discontinuous, or have derivatives that are hard to define. Thus, EAs are naturally suited as optimizers for problems where the output of the system must be determined through experiment or simulation and numerical or experimental noise presents difficulties for finite-difference derivative calculations. As a result, EAs have been used to design applications ranging from antennas to electronic circuits and many EA-generated designs have been shown to outperform human-engineered devices. With these examples in mind, we have provided a simple example of how

an EA can be used to design a lens and we propose the use of EAs for the design of a variety of RF/EO and EO/IR systems.

---

- [1] Driggers, R. G., P. Cox, and T. Edwards. 1999, *Introduction to Infrared and Electro-Optical Systems*, Artech House Inc., Boston.
- [2] Lohn, J. D., G. S. Hornby, and D. S. Linden. 2004, “An Evolved Antenna for Deployment on NASA’s Space Technology 5 Mission,” in U. M. O’Reilly, T. Yu, R. Riolo, and W. Worzel, eds., *Genetic Programming Theory and Practice II*, Kluwer Academic Publishers, Boston, chap. 18.
- [3] Koza, J. R., M. A. Keane, and M. J. Streeter. 2003, “What’s AI Done for Me Lately? Genetic Programming’s Human-Competitive Results,” *IEEE Intelligent Systems*, 18(3):25–31, doi: 10.1109/MIS.2003.1200724.
- [4] Gill, P. E., W. Murray, and M. H. Wright. 1981, *Practical Optimization*, Academic Press, New York.
- [5] Kolda, T. G., R. M. Lewis, and V. Torczon. 2003, “Optimization by Direct Search: New Perspectives on Some Classical and Modern Methods,” *SIAM Review*, 45(3):385–482.
- [6] Kirkpatrick, S., C. D. Gelatt Jr., and M. P. Vecchi. 1983, “Optimization by Simulated Annealing,” *Science*, 220(4598):671–680.
- [7] Bäck, T. 1996, *Evolutionary Algorithms in Theory and Practice: Evolution Strategies, Evolutionary Programming, Genetic Algorithms*, Oxford University Press, New York.
- [8] Press, W. H., S. A. Teukolsky, W. T. Vetterling, and B. P. Flannery. 1992, *Numerical Recipes in C, Second Edition*, Cambridge University Press, New York.
- [9] Al-Sakran, S. H., J. R. Koza, and L. W. Jones. 2005, “Automated Re-Invention of a Previously Patented Optical Lens System Using Genetic Programming,” in M. Keijzer and et al., eds., *Genetic Programming*, Springer-Verlag, Berlin, vol. 3447 of *Lecture Notes in Computer Science*, doi: 10.1007/b107383.
- [10] Storn, R. and R. Price. 1997, “Differential Evolution - A Simple and Efficient Heuristic for Global Optimization Over Continuous Spaces,” *Journal of Global Optimization*, 11(4):341–359.
- [11] Koza, J. R., F. H. Bennett III, D. Andre, and M. A. Keane. 1999, *Genetic Programming III: Darwinian Invention and Problem Solving*, Morgan Kaufmann, San Francisco.
- [12] Koza, J. R., M. A. Keane, and M. J. Streeter. 2003, “Evolving Inventions,” *Scientific American*, 288(2):52–59.
- [13] Gopalakrishnan, G. K., R. P. Moeller, M. M. Howerton, W. K. Burns, K. J. Williams, and R. D. Esman. 1995, “A low-loss downconverting analog fiber-optic link,” *IEEE Trans. Micro. Theory Techn.*, 43(9).
- [14] Williams, K. J. and R. D. Esman. 1998, “Optically Amplified Downconverting Link with Shot-Noise-Limited Performance,” *IEEE Photonics Technology Letters*, 8(1).

[15] Ferreira, C. 2001, “Gene Expression Programming: A New Adaptive Algorithm for Solving Problems,” *Complex Systems*, 13(2):87–129.

